



Document technique sur l'installation d'une machine virtuelle

Par

Grégoire LE BARON

Table des matières :

1- Introduction

2- Prérequis

3- Installation

- a. Création d'une machine virtuelle**
- b. Paramétrage de la machine virtuelle**
- c. Paramétrage de l'OS Lubuntu**

4- Réalisation d'un script bash

1- Introduction

Dans le cadre de notre cursus scolaire, nous sommes amenés à travailler sur plusieurs systèmes d'exploitation afin de nous y familiariser.

2- Prérequis

Dans ce document technique, nous allons utiliser le logiciel Oracle VM VirtualBox qui est un hyperviseur de type 2 open source. Un hyperviseur de type 2, c'est-à-dire qu'il s'exécute sur un système d'exploitation déjà existant.

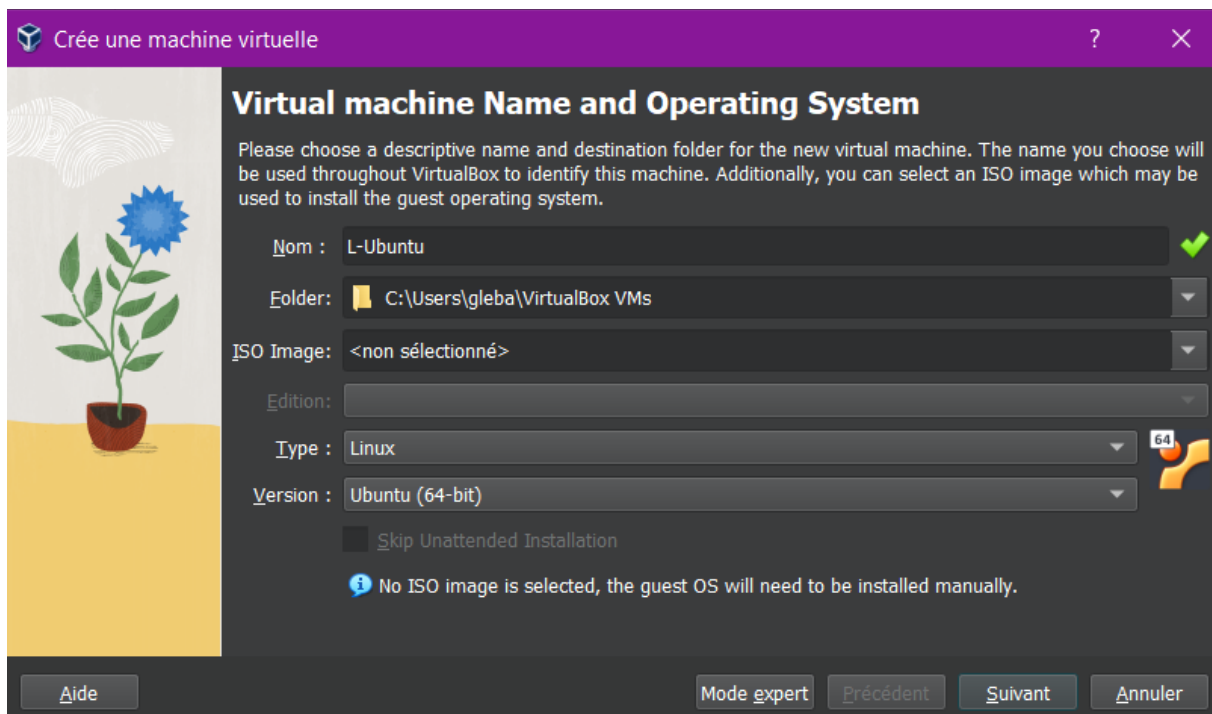
Ensuite, pour la machine virtuelle, nous avons besoin d'une image ISO d'un système d'exploitation qui correspond à une copie identique à l'originale. Dans notre cas, nous allons utiliser une image ISO du système d'exploitation Lubuntu.

Enfin, d'un point de vue Hardware, il est nécessaire d'avoir minimum 4Go de mémoire RAM disponible, 1 cœur de processeur ainsi que 25 Go de mémoire.

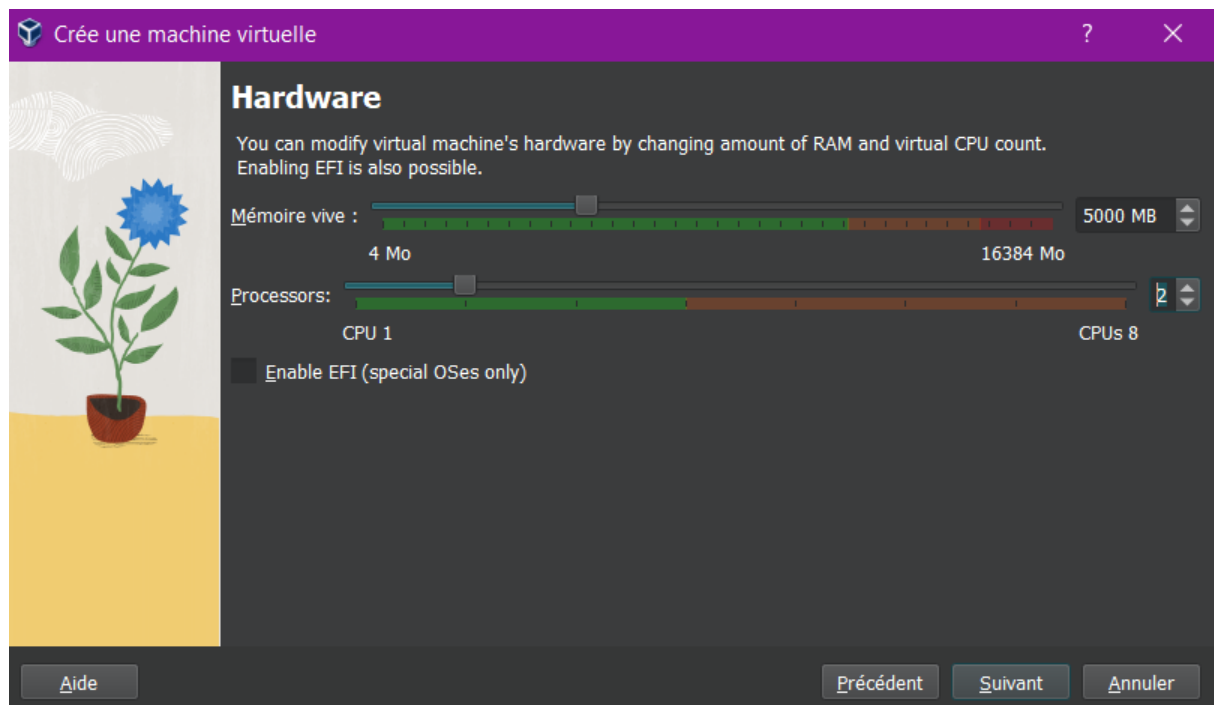
3- Installation

a. Création d'une machine virtuelle

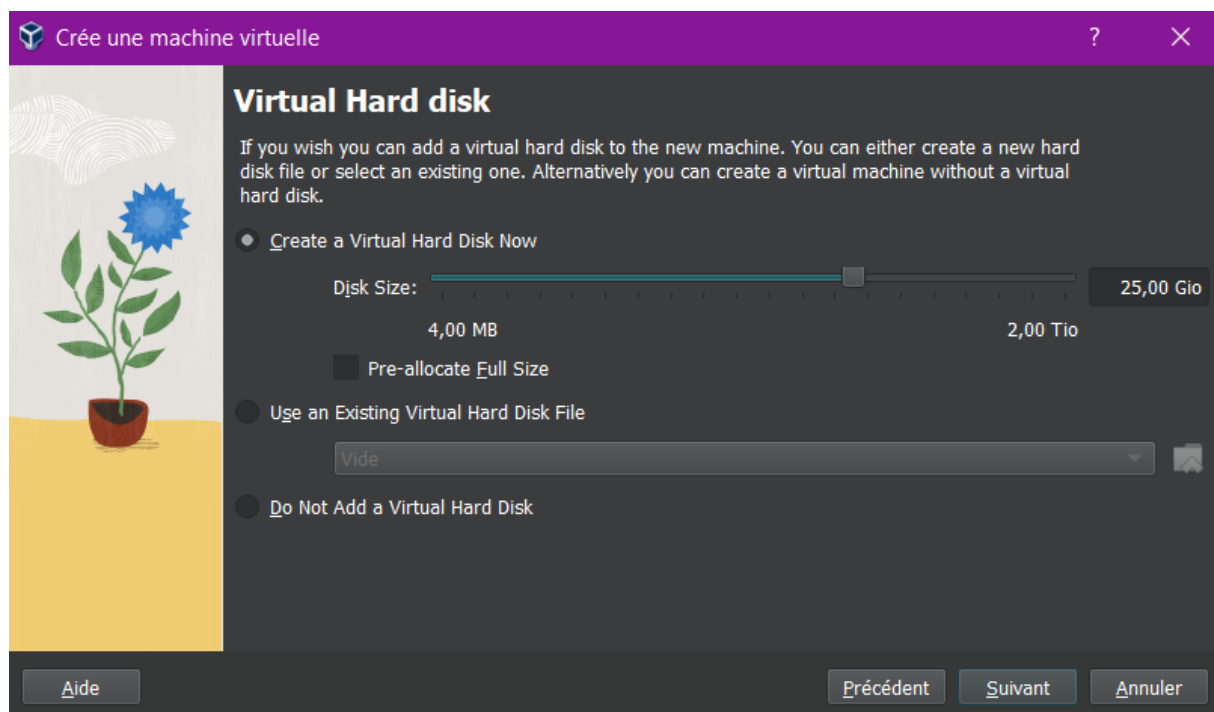
On installe une machine virtuelle basé sur l'OS Lubuntu.



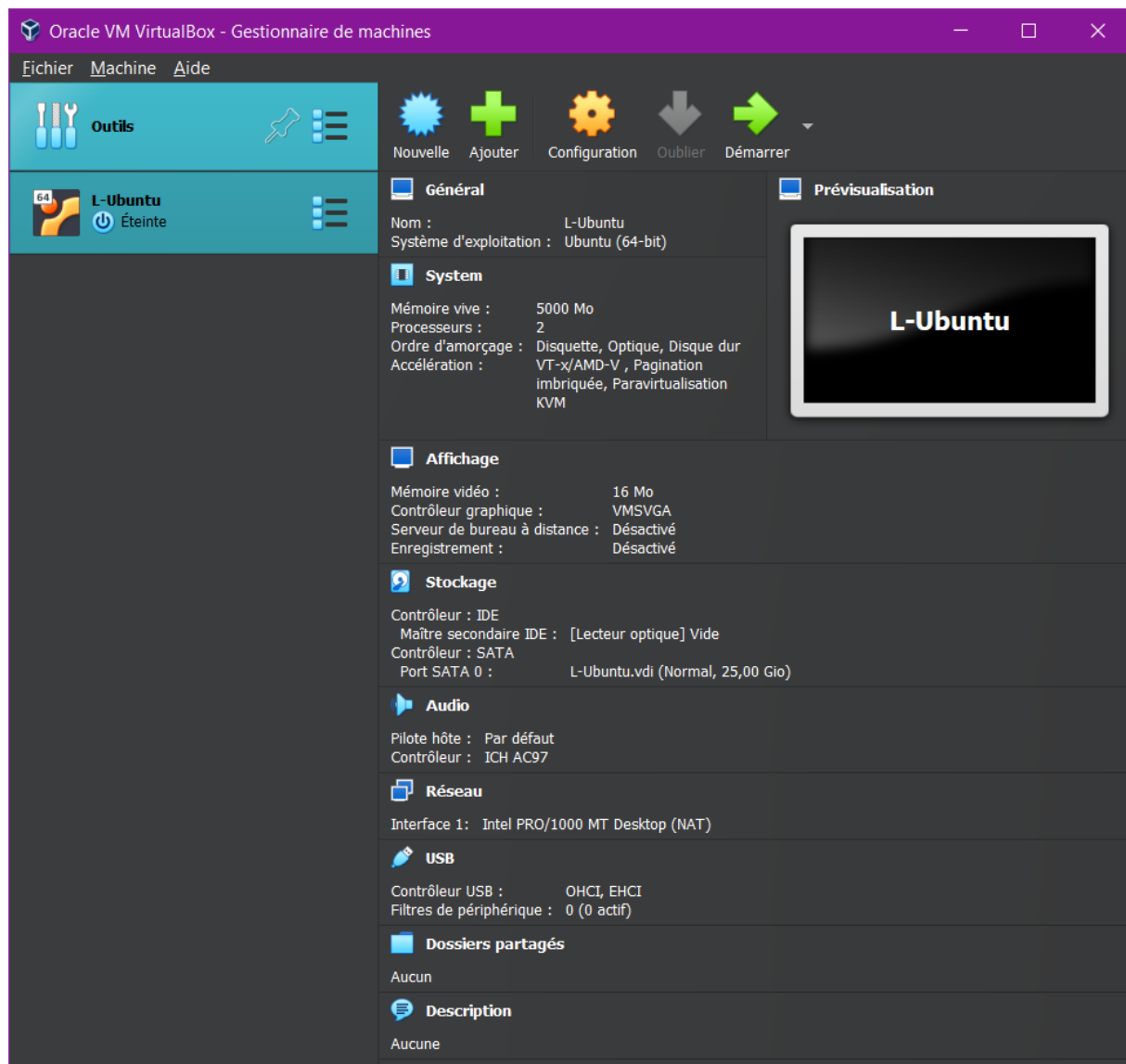
Après le lancement du logiciel, nous créons une nouvelle VirtualBox. Pour cela, nous cliquons sur l'onglet « Nouvelle » qui nous ouvre la fenêtre ci-dessus. Nous y rentrons le nom de la machine virtuelle, le noyau (Linux) et le système d'exploitation (Ubuntu).



Ensuite, nous paramétrons les ressources que la machine virtuelle peut allouer sur l'hardware. Dans notre situation, 5 Go de RAM et 2 cœurs du processeur.

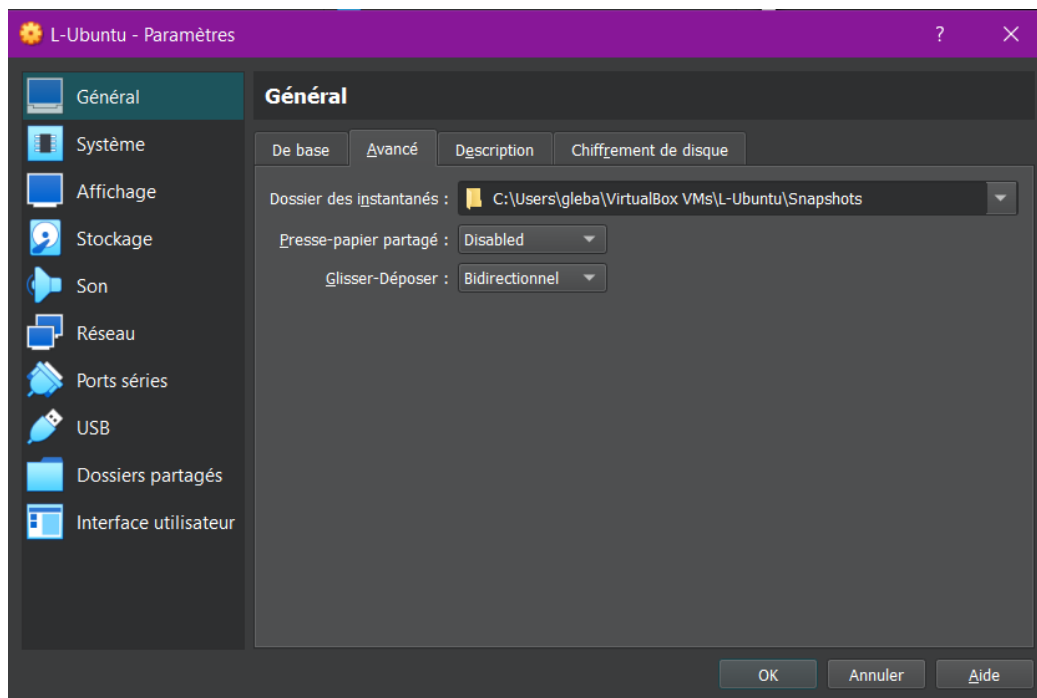


Enfin, nous paramétrons la mémoire sur le disque dure dédiée à la machine virtuelle.

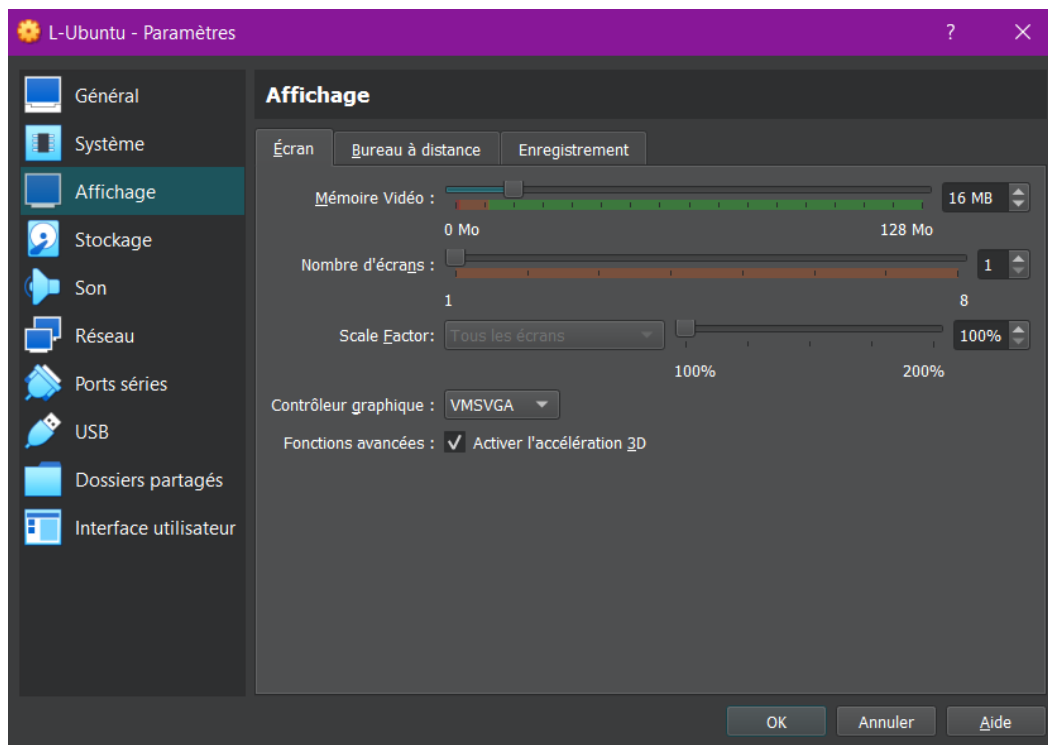


Notre machine virtuelle est prête, nous le retrouvons dans la liste de toutes les machines virtuelles. Sur la partie droite, nous retrouvons ses caractéristiques que nous avons préalablement établi.

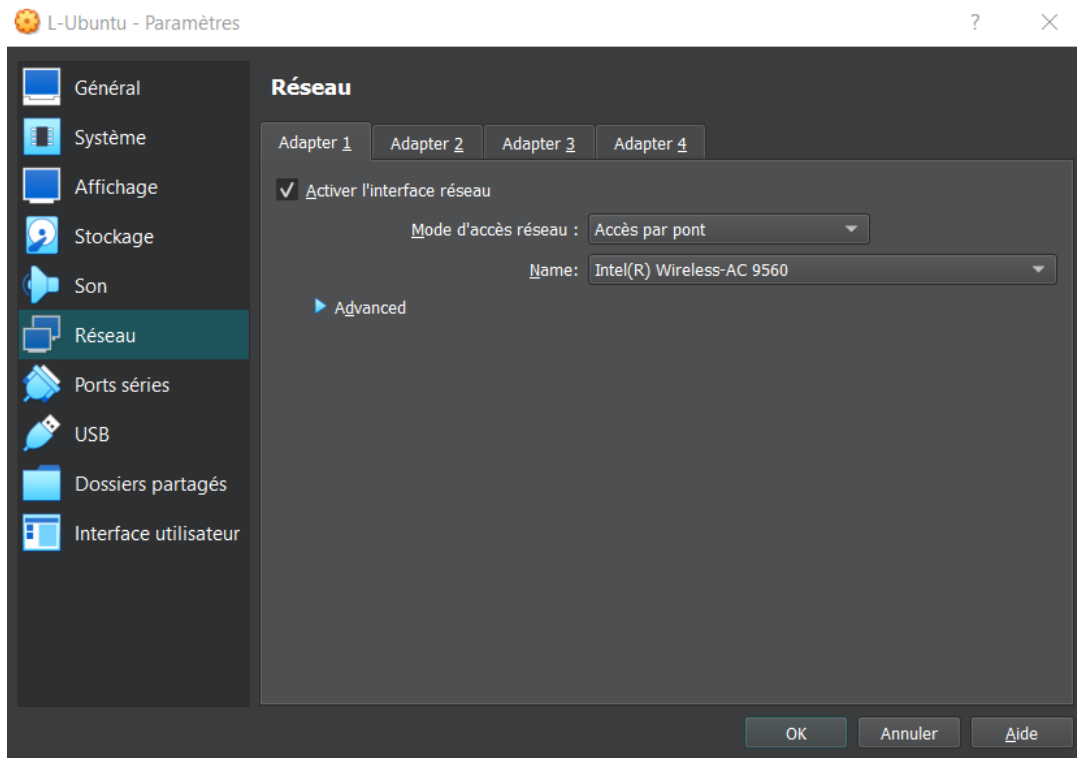
b. Paramétrage de la machine virtuelle



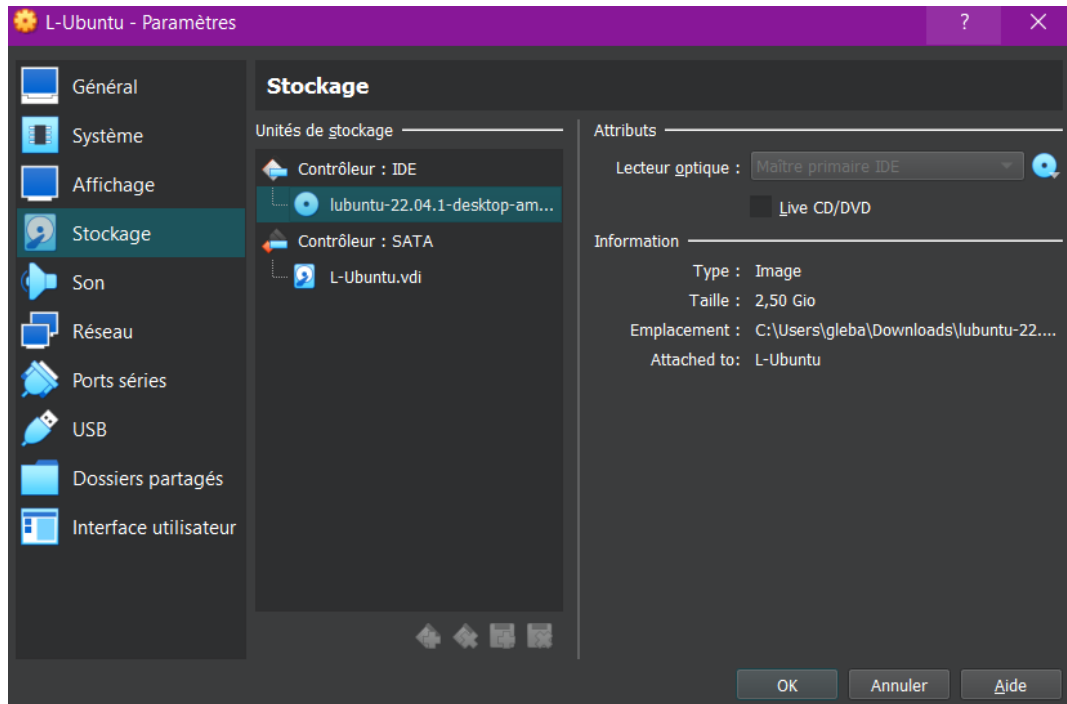
Afin d'accéder à l'option copier-coller, nous modifions le paramètre « Glisser-Déposer » se trouvant dans la catégorie « Avancé » des paramètres généraux. Nous passons ce paramètre, qui est à l'origine sous forme « Disabled », en « Bidirectionnel ».



Ensuite, j'active le paramètre « Fonctions avancées : Activer l'accélération 3D » situé dans l'onglet « Affichage ».

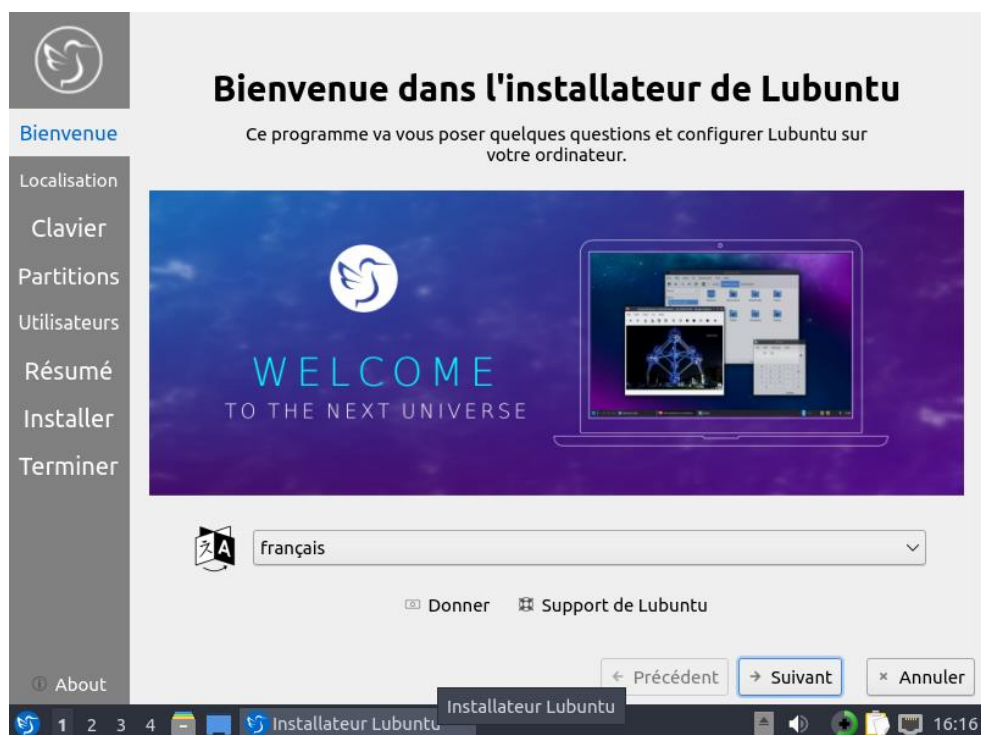


Je sélectionne l'option « Accès par pont » pour le paramètre « Mode d'accès réseau » situé dans l'onglet « Réseau ».

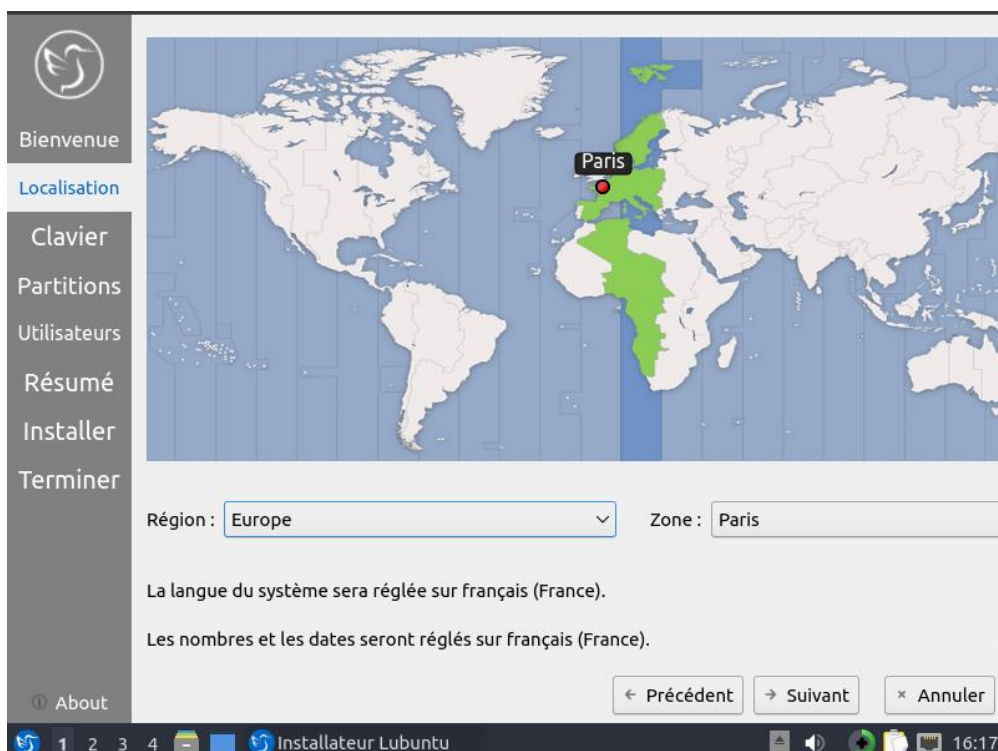


Sélectionner dans « Attributs » pour le paramètre « lecteur optique » : Maître Primaire IDE. Ensuite sur le disque bleu qui apparaît sur sa droite, sélectionné le fichier « lubuntu-22.04.1-desktop-amd64 ». Appuyez sur « ok ».

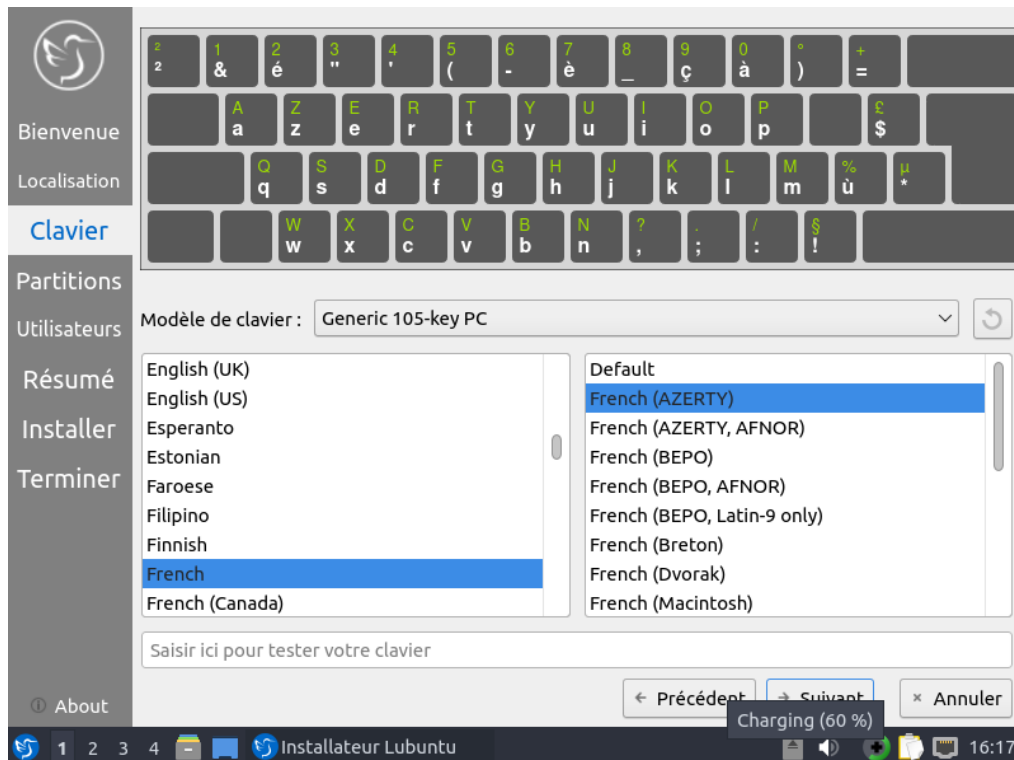
c. Paramétrage de l'OS Ubuntu



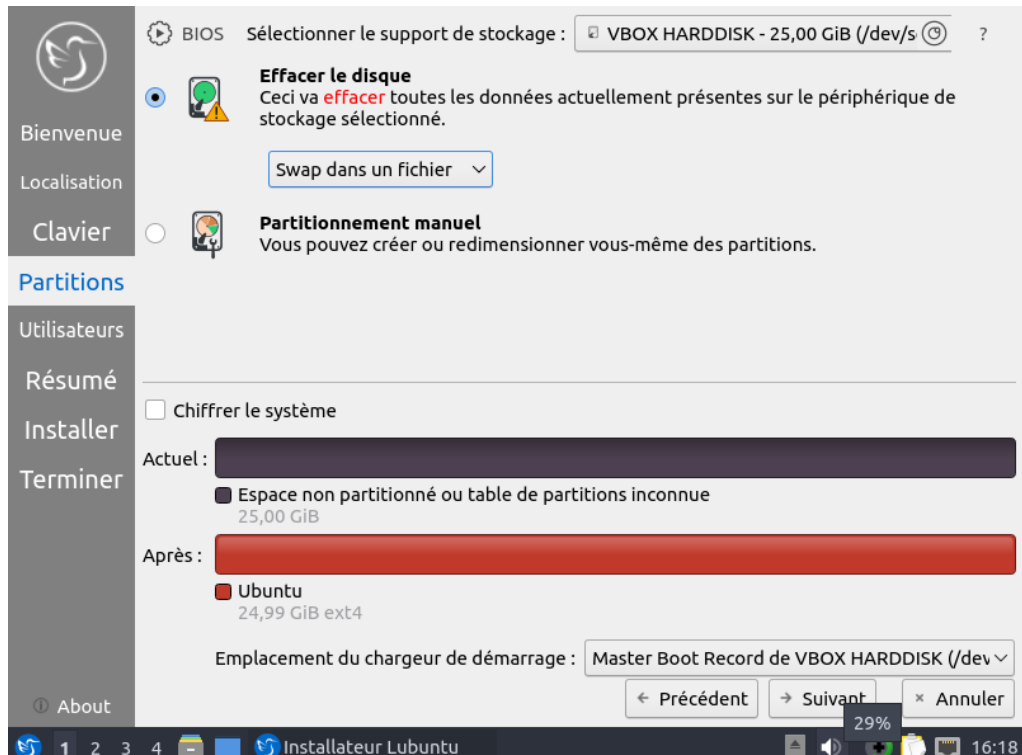
Maintenant, nous ouvrons l'application « Installer Ubuntu », ensuite nous sélectionnons la langue, ici « français ». Puis, nous appuyons sur « suivant ».



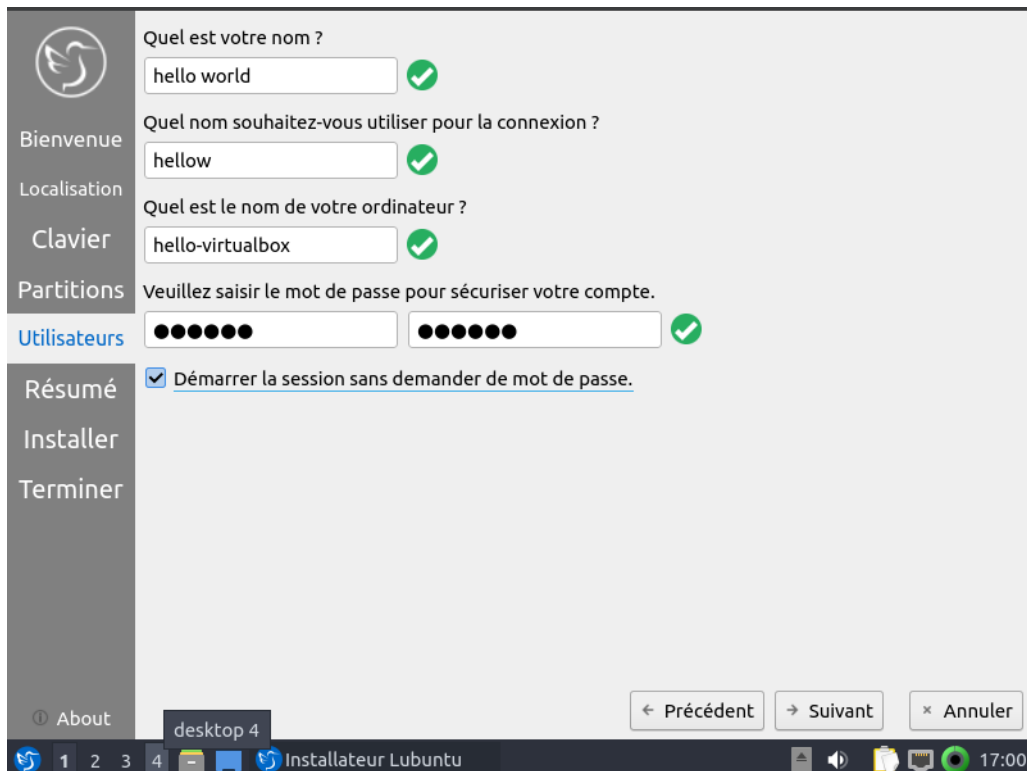
Ensuite, nous sélectionnons la région « Europe » et zone « Paris ». Puis, nous appuyons sur « suivant ».



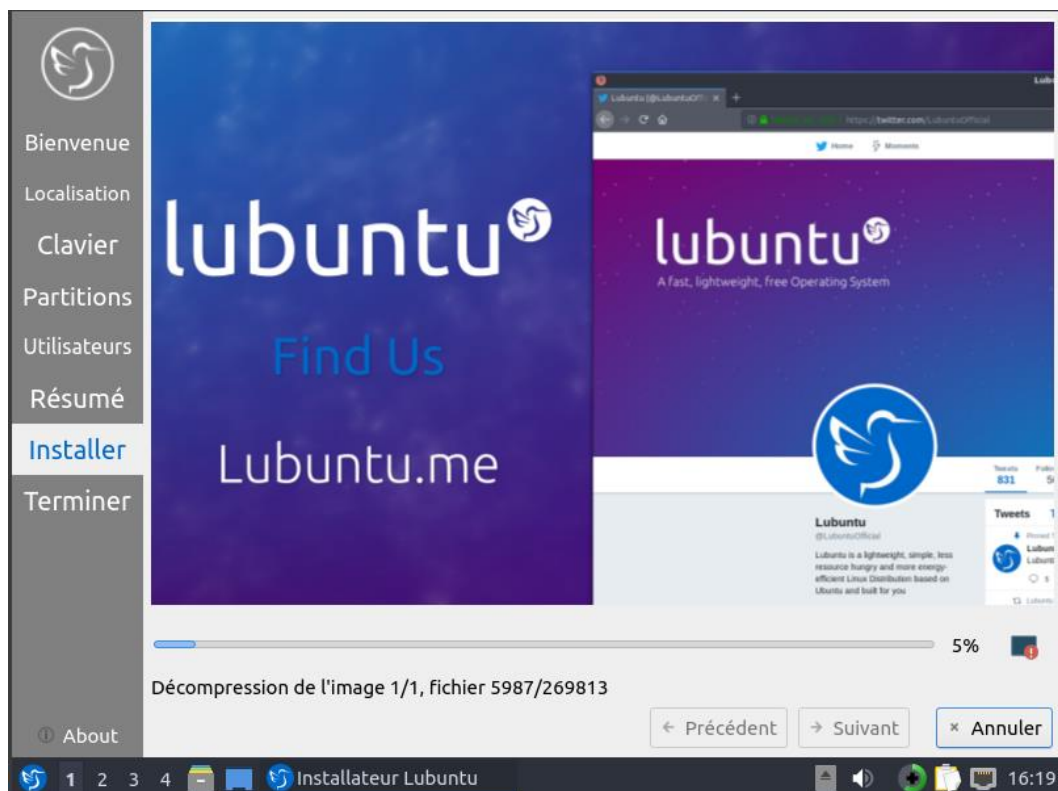
Nous paramétrons notre clavier. Pour cela, nous sélectionnons « French » dans la colonne de gauche et « French (AZERTY) » dans la colonne de droite. Puis, nous appuyons sur « suivant ».



Ensuite, nous sélectionnons « Effacer le disque ». Puis, nous appuyons sur « suivant ».

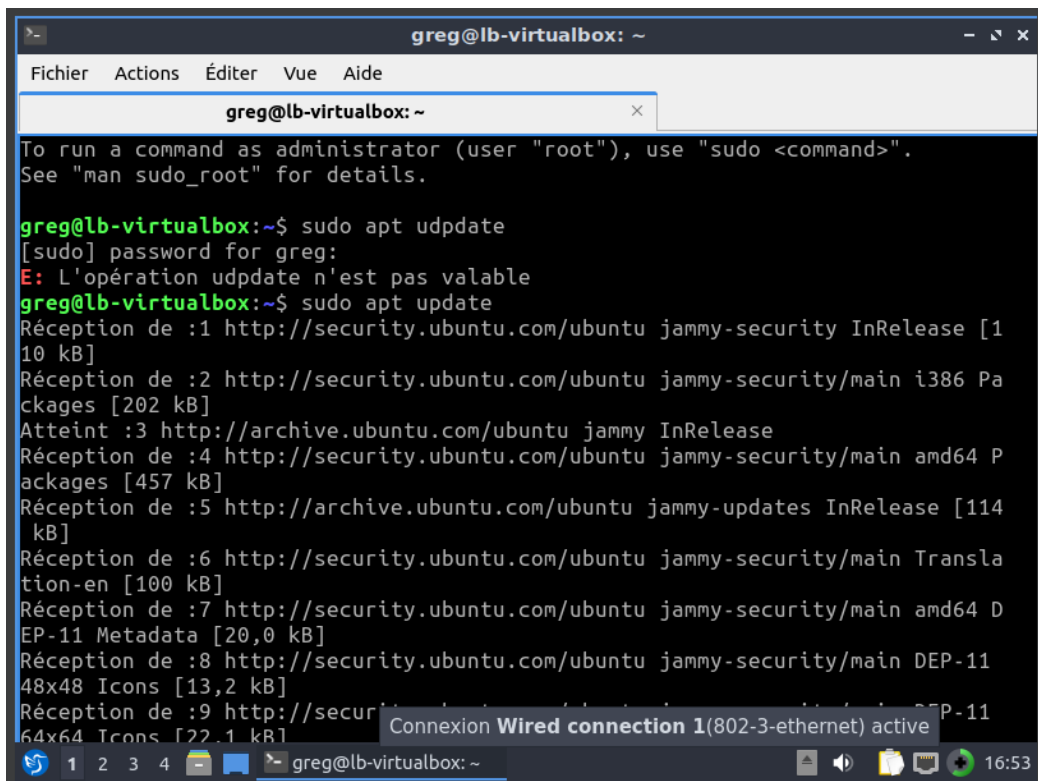


Par la suite, nous créons un profil. Puis, nous appuyons sur « suivant ».



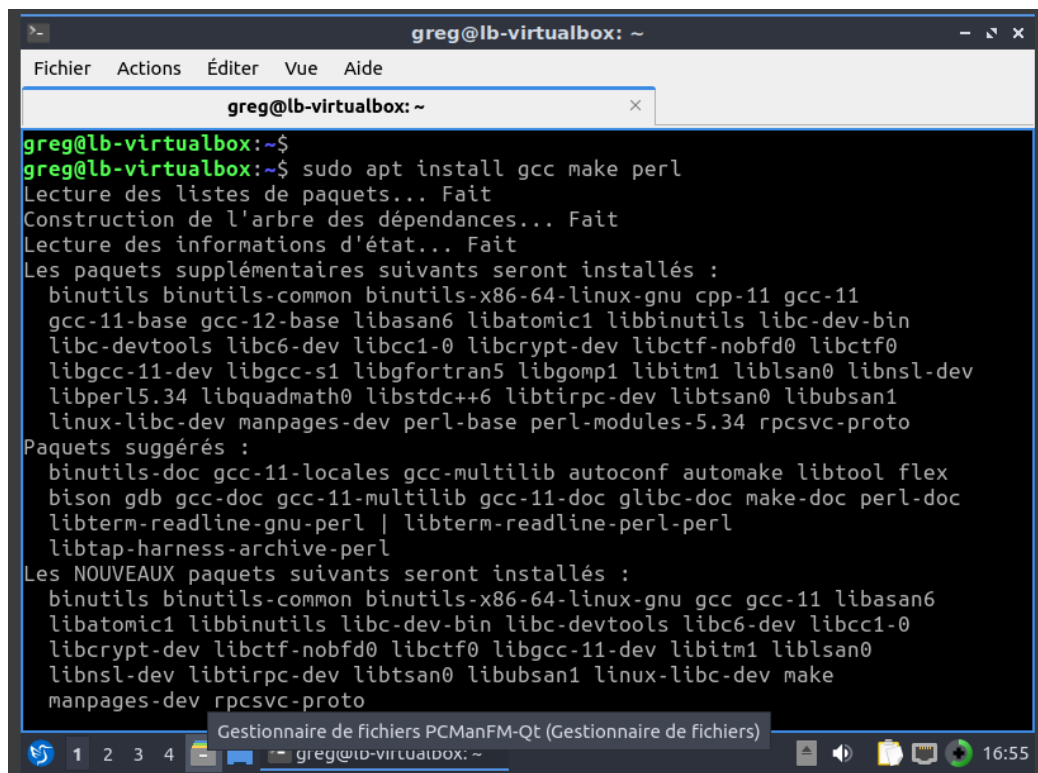
Ci-dessus, l'image du téléchargement avant de redémarrer notre machine virtuelle.

Maintenant, nous réalisons les mises à jour de l'OS.



```
greg@lb-virtualbox: ~  
Fichier Actions Éditer Vue Aide  
greg@lb-virtualbox: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
greg@lb-virtualbox:~$ sudo apt update  
[sudo] password for greg:  
E: L'opération update n'est pas valable  
greg@lb-virtualbox:~$ sudo apt update  
Réception de :1 http://security.ubuntu.com/ubuntu jammy-security InRelease [1  
10 kB]  
Réception de :2 http://security.ubuntu.com/ubuntu jammy-security/main i386 Pa  
ckages [202 kB]  
Atteint :3 http://archive.ubuntu.com/ubuntu jammy InRelease  
Réception de :4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 P  
ackages [457 kB]  
Réception de :5 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [114  
kB]  
Réception de :6 http://security.ubuntu.com/ubuntu jammy-security/main Transla  
tion-en [100 kB]  
Réception de :7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 D  
EP-11 Metadata [20,0 kB]  
Réception de :8 http://security.ubuntu.com/ubuntu jammy-security/main DEP-11  
48x48 Icons [13,2 kB]  
Réception de :9 http://security.ubuntu.com/ubuntu jammy-security/main DEP-11  
64x64 Icons [22,1 kB]  
Connexion Wired connection 1(802-3-ethernet) active  
greg@lb-virtualbox: ~
```

Nous entrons la commande « `sudo apt update` » pour mettre à jour le système.



```
greg@lb-virtualbox: ~$  
greg@lb-virtualbox:~$ sudo apt install gcc make perl  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Les paquets supplémentaires suivants seront installés :  
  binutils binutils-common binutils-x86-64-linux-gnu cpp-11 gcc-11  
  gcc-11-base gcc-12-base libasan6 libatomic1 libbinutils libc-dev-bin  
  libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0  
  libgcc-11-dev libgcc-s1 libgfortran5 libgomp1 libitm1 liblsan0 libnsd-dev  
  libperl5.34 libquadmath0 libstdc++6 libtirpc-dev libtsan0 libubsan1  
  linux-libc-dev manpages-dev perl-base perl-modules-5.34 rpcsvc-proto  
Paquets suggérés :  
  binutils-doc gcc-11-locales gcc-multilib autoconf automake libtool flex  
  bison gdb gcc-doc gcc-11-multilib gcc-11-doc glibc-doc make-doc perl-doc  
  libterm-readline-gnu-perl | libterm-readline-perl-perl  
  libtap-harness-archive-perl  
Les NOUVEAUX paquets suivants seront installés :  
  binutils binutils-common binutils-x86-64-linux-gnu gcc gcc-11 libasan6  
  libatomic1 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0  
  libcrypt-dev libctf-nobfd0 libctf0 libgcc-11-dev libitm1 liblsan0  
  libnsd-dev libtirpc-dev libtsan0 libubsan1 linux-libc-dev make  
  manpages-dev rpcsvc-proto  
Gestionnaire de fichiers PCManFM-Qt (Gestionnaire de fichiers)  
greg@lb-virtualbox: ~
```

Nous entrons la commande « `sudo apt install gcc make perl` »

```

1) ...
greg@lb-virtualbox:~$ cd /media/greg/VBox_GAs_7.0.2/
cert/ NT3x/ OS2/
greg@lb-virtualbox:~$ cd /media/greg/VBox_GAs_7.0.2/
greg@lb-virtualbox:/media/greg/VBox_GAs_7.0.2$ sudo ./
autorun.sh                                VBoxDarwinAdditionsUninstall.tool
cert/                                    VBoxLinuxAdditions.run
NT3x/                                    VBoxWindowsAdditions-amd64.exe
OS2/                                    VBoxWindowsAdditions.exe
runasroot.sh                            VBoxWindowsAdditions-x86.exe
greg@lb-virtualbox:/media/greg/VBox_GAs_7.0.2$ sudo ./VBoxLinuxAdditions.run
Verifying archive integrity... 100% MD5 checksums are OK. All good.
Uncompressing VirtualBox 7.0.2 Guest Additions for Linux 100%
VirtualBox Guest Additions installer
VirtualBox Guest Additions: Starting.
VirtualBox Guest Additions: Setting up modules
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel
modules. This may take a while.
VirtualBox Guest Additions: To build modules for other installed kernels, run
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup <version>
VirtualBox Guest Additions: or
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup all
VirtualBox Guest Additions: Building the modules for kernel 5.15.0-43-generic
.
update-initramfs: Generating /boot/initrd.img-5.15.0-43-generic
VirtualBox Guest Additions: Running kernel modules will not be replaced until
the system is restarted
greg@lb-virtualbox:/media/greg/VBox_GAs_7.0.2$ █

```

La première commande « `cd /media/greg/VBox_GAs_7.0.2` » nous permet d'ouvrir un le fichier, ensuite la commande « `sudo ./VBoxLinuxAdditions.run` » qui nous permet d'avoir la machine virtuelle en plein écran.

4- Réalisation d'un script bash

Nous allons maintenant écrire un script qui permet de créer un fichier en demandant à l'utilisateur le nom du fichier et qui contient « Hello Word ».

Pour créer un script, nous exécutons la commande « `nano "nom du fichier.sh"` ». Cela nous ouvre une fenêtre dans lequel nous écrivons le script. Puis nous écrivons l'algorithme suivant :

```

GNU nano 6.2
#!/bin/bash

echo "Quel est le nom du fichier que vous souhaitez créer ?"

# Enregistre le nom du fichier donné par l'utilisateur dans la variable a
read a

# Créer un fichier avec comme nom la variable a
touch $a

# Enregistre dans le fichier "Hello World"
echo "Hello World" > $a

```

Afin d'exécuter ce script, nous rentrons la commande « `bash « bash "nom du fichier.sh" »` ». Nous vérifions que le script fonctionne :

```
greg@lb-virtualbox:~$ nano script.sh
greg@lb-virtualbox:~$ ls
Desktop Documents Images Modèles Musique nano.15363.save Public script.sh snap Téléchargements Vidéos
greg@lb-virtualbox:~$ bash script.sh
Quel est le nom du fichier que vous souhaitez créer ?
unfichier
greg@lb-virtualbox:~$ ls
Desktop Documents Images Modèles Musique nano.15363.save Public script.sh snap Téléchargements unfichier Vidéos
greg@lb-virtualbox:~$ cat unfichier
Hello World
greg@lb-virtualbox:~$
```

Nous observons bien qu'un fichier ayant comme nom « unfichier » a été créé et contient « Hello Word ».